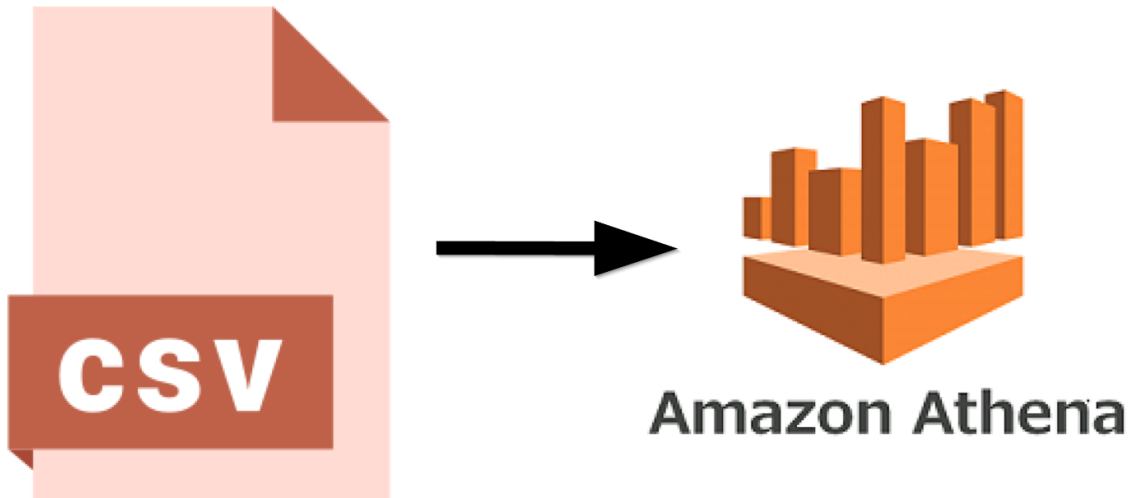


Load a CSV file into AWS Athena for SQL Analysis



This tutorial will teach you how to load a CSV file into AWS Athena so that you can analyze it using SQL queries. This process will work for any CSV file, but in this tutorial we'll be using the free version of our World Cities Database (<https://simplemaps.com/data/world-cities>). You could also use it with any of our many geographic databases (<https://simplemaps.com/data>). Let's get started

1. Login or create an AWS Account (<https://aws.amazon.com>).
2. Download the Basic World Cities Database (<https://simplemaps.com/data/world-cities>).
This is free and won't require you to provide any information. However, attribution is required. After downloading, unzip the folder. We'll be using the `worldcities.csv` file.

Download Database

We've decided to make this version of the database available to anyone, free of charge. If you find valuable, please consider spreading the word:

- With this [prepopulated tweet](#) on Twitter **1.**
- By attributing our work with a backlink to <https://simplemaps.com/data/world-cities>. **2.**

Spreading the word on social media is optional, but the basic database [license](#) does require you to your source in some fashion.

Thanks!

[Download](#)

refund policy	N/A	30-day guarantee
One-time fee	Free	\$199
	Download	Buy Now!

3. Clean and format the CSV as needed. The CSV file should be encoded as UTF-8. Due to the way Athena parses CSV files, we also need to remove all quotes from the CSV file. The easiest way to do this is to open the CSV file in LibreOffice (<https://www.libreoffice.org/download/download-libreoffice/>):

ImportCharacter set: **Unicode (UTF-8)**Language: **Default - English (USA)**From row: **1****Separator Options** Fixed width Separated by Tab Comma Semicolon Space Other Merge delimiters Trim spaces

String delimiter: "

Other Options Format quoted field as text Detect special numbers**Fields**Column type:

	Standard	Standard	Standard	Standard	St
1	city	city_ascii	lat	lng	co
2	Tokyo	Tokyo	35.6839	139.7744	Ja
3	Jakarta	Jakarta	-6.2146	106.8451	Id
4	Delhi	Delhi	28.6667	77.2167	Id
5	Manila	Manila	14.6000	120.9833	Ph
6	São Paulo	Sao Paulo	-23.5504	-46.6339	Br
7	Seoul	Seoul	37.5600	126.9900	So
8	Mumbai	Mumbai	19.0758	72.8775	Id
9	Shanghai	Shanghai	31.1667	121.4667	Ch

Help**O**K**C**ancel

and then save it. Note the absence of quotation marks.

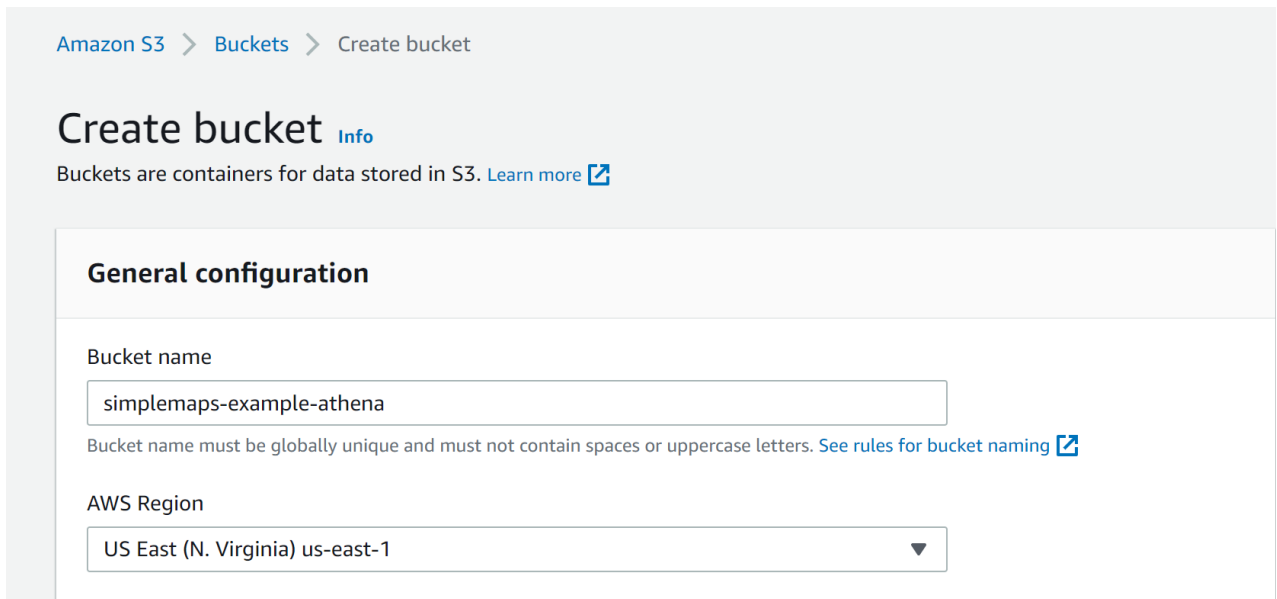
```

1 city,city_ascii,lat,lng,country,iso2,iso3,admin_name,capital,population,id
2 Tokyo,Tokyo,35.6839,139.7744,Japan,JP,JPN,Tōkyō,primary,39105000,1392685764
3 Jakarta,Jakarta,-6.2146,106.8451,Indonesia,ID,IDN,Jakarta,primary,35362000,1360771077
4 Delhi,Delhi,28.6667,77.2167,India,IN,IND,Delhi,admin,31870000,1356872604
5 Manila,Manila,14.6,120.9833,Philippines,PH,PHL,Manila,primary,23971000,1608618140
6 São Paulo,Sao Paulo,-23.5504,-46.6339,Brazil,BR,BRA,São Paulo,admin,22495000,1076532519
7 Seoul,Seoul,37.56,126.99,South Korea,KR,KOR,Seoul,primary,22394000,1410836482
8 Mumbai,Mumbai,19.0758,72.8775,India,IN,IND,Mahārāshtra,admin,22186000,1356226629
9 Shanghai,Shanghai,31.1667,121.4667,China,CN,CHN,Shanghai,admin,22118000,1156073548
10 Mexico City,Mexico City,19.4333,-99.1333,Mexico,MX,MEX,Ciudad de México,primary,21505000,1484247881
11 Guangzhou,Guangzhou,23.1288,113.259,China,CN,CHN,Guangdong,admin,21489000,1156237133
12 Cairo,Cairo,30.0444,31.2358,Egypt,EG,EGY,Al Qāhirah,primary,19787000,1818253931
13 Beijing,Beijing,39.904,116.4075,China,CN,CHN,Beijing,primary,19437000,1156228865
14 New York,New York,40.6943,-73.9249,United States,US,USA,New York,,18713220,1840034016

```

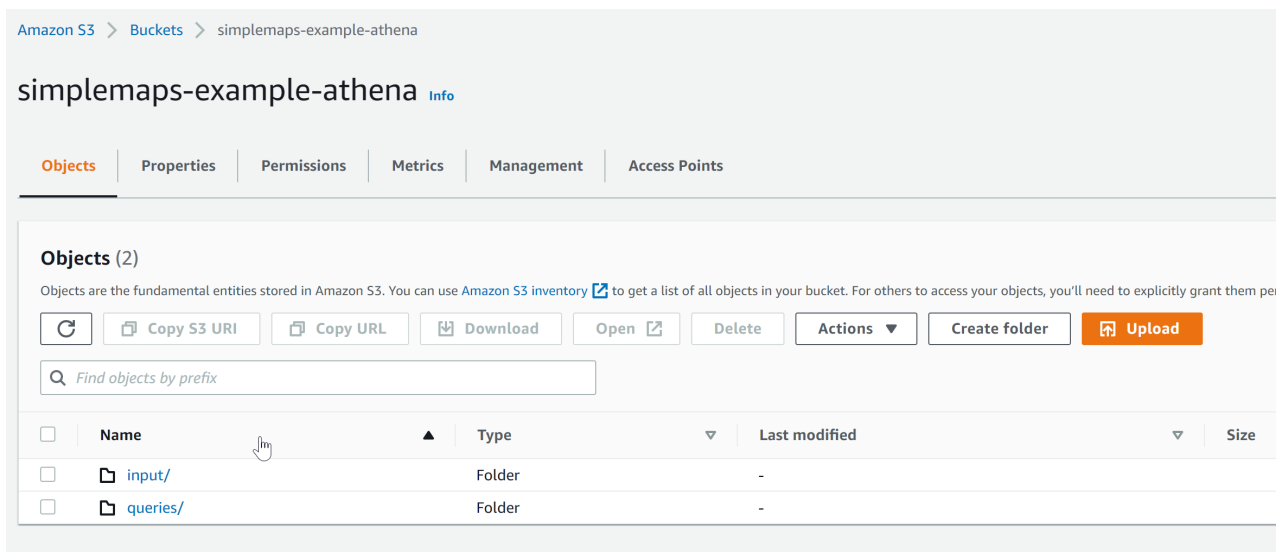
If you just want the CSV file after this has been done, you can download it here (/static/img/resources/csv-to-athena/worldcities.csv). Note: Do not try to do this with Excel because it does not handle UTF-8 characters well.

4. Create a new AWS S3 Bucket. We'll use the name `simplemaps-example-athena`. You will need to use a unique name of your own.



Use all of the default settings.

5. Create two folders in the bucket: `input` and `queries`. Athena will use the `queries` folder to store the queries it runs.
6. Upload the `worldcities.csv` file to the `input` folder you just created.
7. If you did things correctly, your bucket should look like this:



<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	input/	Folder	-	
<input type="checkbox"/>	queries/	Folder	-	

and the input folder should look like this:

Amazon S3 > Buckets > simplemaps-example-athena > input/

input/

Objects | Properties

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your

Copy S3 URI Copy URL Download Open Delete Actions Create

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	worldcities.csv	csv	September 14, 2022, 13:01:51 (UTC-04:00)

8. Open Amazon Athena and click **Explore the Query Editor**.

Amazon Athena

Start querying data instantly.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 and other federated data sources using standard SQL.

Begin querying your data

Discover the query editor and start right away.

[Explore the query editor](#)

Pricing

N. Virginia - Data scanned

[Cost calculator](#)

How it works

9. Go to the Settings tab and set the location of the query result to be `s3://simplemaps-example-athena/queries`. Except, of course, use your own unique bucket name.

Manage settings

Query result location and encryption

Location of query result - *optional*

Enter an S3 prefix in the current region where the query result will be saved as an object.

[View](#)[Browse S3](#)

Expected bucket owner - *optional*

Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Assign bucket owner full control over query results

Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

Encrypt query results

[Cancel](#)[Save](#)

This is where Athena will store the queries that it runs.

10. It's possible to import the data using the "Create" button, but we will import the data using a Query. Paste the following into the query area:

```

CREATE EXTERNAL TABLE IF NOT EXISTS `default`.`worldcities` (
  `city` string,
  `city_ascii` string,
  `lat` double,
  `lng` double,
  `country` string,
  `iso2` string,
  `iso3` string,
  `admin_name` string,
  `capital` string,
  `population` int,
  `id` int
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = ',',
  'field.delim' = ','
)
LOCATION 's3://simplemaps-example-athena/input/'
TBLPROPERTIES ('skip.header.line.count'='1')

```

then click Run:

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings | Workgroup: primary

Data

Data source: Aws **Alternative way to import data**

Database: default

Tables and views: **Create**

Filter tables and views

Tables (0) | Views (0)

Query 1

```

1 CREATE EXTERNAL TABLE IF NOT EXISTS `default`.`worldcities` (
2   `city` `string`,
3   `city_ascii` `string`,
4   `lat` `double`,
5   `lng` `double`,
6   `country` `string`,
7   `iso2` `string`,
8   `iso3` `string`,
9   `admin_name` `string`,
10  `capital` `string`,
11  `population` `int`,
12  `id` `int`
13 )
14 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
15 WITH SERDEPROPERTIES (

```

SQL Ln 21, Col 2

Run again | Explain | Cancel | Save | Clear | Create

A few things to note on what this does:

- We have defined all of the fields in the dataset and given them an appropriate type.
- We've told Athena to use the LazySimpleSerDe (<https://docs.aws.amazon.com/athena/latest/ug/lazy-simple-serde.html>) CSV parser. We use this parser because it allows Null values for numbers. It does not support quoted values, though, which is why we removed those earlier with LibreOffice. You can learn more about parsers in this helpful guide (<https://athena.guide/articles/working-with-csv/>).
- We've told the parser that fields are delimited by commas and that the first row contains fieldnames which can be skipped.
- We've specified the location of the CSV file. We only need to provide the folder, not the file itself.

11. If the import was successful, you'll see a green "Completed" message:

Query results | Query stats

Completed | Time in queue: 85 ms

Query successful.

12. Test the data with the following query:

```
select city, population from worldcities where iso2='US' order by population desc limit
```

```
1 select city, population from worldcities where iso2='US' order by population desc limi
```

and you should get the five largest cities in the United States:

Query results | Query stats

✓ Completed Time in queue: 177 ms Run time: 522 ms Data scanned: 3.

Results (5) Copy Download results

Search rows < 1 >

#	city	population
1	New York	18713220
2	Los Angeles	12750807
3	Chicago	8604203
4	Miami	6445545
5	Dallas	5743938

13. Now you can query your CSV file using any other SQL queries you like!

14. To learn how to run these queries over a url, see our next tutorial [Query an Athena database over a URL with AWS Lambda \(/resources/athena-over-url\)](#).

Resources

Tools, freebies, and articles about mapping and web development. Things we've learned and want to share!

- [Introduction \(./\)](#)

Free SVG Maps

- [All Free SVG Maps \(svg-maps\)](#)
- [World SVG Map \(svg-world\)](#)
- [USA SVG Map \(svg-us\)](#)

Free Interactive Maps

- [Free World Map \(free-world-map\)](#)
- [Free US Map \(free-us-map\)](#)
- [Free Continent \(free-continent-map\)](#)

Data Articles

- [CSV to Athena \(/resources/csv-to-athena\)](#)
- [Athena over URL \(/resources/athena-over-url\)](#)
- [Import CSV into MySQL \(/resources/import-csv-mysql\)](#)
- [Import data subset with Power Query \(/resources/power-query\)](#)
- [Retrieve demographic data with VLOOKUP \(/resources/vlookup\)](#)
- [Import CSV into Tableau \(/resources/import-tableau\)](#)
- [Visualize places in QGIS \(/resources/cities-qgis\)](#)
- [Filter places in Excel \(/resources/filter-excel\)](#)
- [Location distance code \(.js and .py\) \(/resources/location-distance\)](#)

Map Articles

- [Guide to MapShaper \(/resources/guide-to-mapshaper\)](#)
- [Overlay polygons on Google Maps \(/resources/shapes-google-maps\)](#)
- [Using Google Cloud \(/resources/google-cloud-project\)](#)
- [Introduction to customizing SVG \(/resources/customize-us-svg\)](#)
- [Software we use at SimpleMaps \(/resources/software-at-simplemaps\)](#)
- [Alternative to Google Maps \(/resources/alternative-to-google-maps\)](#)
- [Flash vs. HTML5 maps \(/resources/flash-vs-html5\)](#)

Data and Tools

- [US Address/Cities Geocoder \(us-geocoder\)](#)
- [World cities data \(by country\) \(/resources/free-country-cities\)](#)

Visualizations

- [US Gold Medalists Rio 2016 \(/viz/rio16\)](#)
- [US Unemployment \(auto-updated\) \(/viz/unemployment\)](#)
- [Snakes and Bears \(/viz/wild\)](#)
- [Zip Code Visualizations \(/resources/zip-code-visualizations\)](#)

Links

- [Raphael.js \(http://dmitrybaranovskiy.github.io/raphael/\)](#)
- [MapShaper \(http://www.mapshaper.org/\)](#)
- [Old articles about Flash maps \(/flash\)](#)